

論文96-33A-11-15

빌딩블록의 레이아웃 설계를 위한 계층적 배치 방법

(A Hierarchical Placement Method for Building Block Layout Design)

姜秉益*, 李建培**

(Byung-Ik Kang and Keon-Bae Lee)

요약

본 논문에서는 크기와 모양이 미리 결정되어 있는 사각형 블록의 배치 문제를 풀기 위한 새로운 배치 알고리즘을 제안한다. 제안한 방법은 크기와 모양이 서로 다른 여러 개의 사각형 블록을 계층적으로 배치함으로써 칩 면적을 최소화하는 방법이다. 배치 방법은 크게 계층적 분할, 면적/모양 예측, 계층적 패턴 배치, 중첩 제거, 모듈 회전 의 과정으로 구성된다. 회로를 계층적으로 분할하여 계층 트리를 구성한 후, 분할결과와 각 모듈의 핀 정보를 이용하여 배치가 수행되기 전 단계에서 계층적으로 배선영역과 모듈의 모양을 예측한다. 또한 모듈의 상대적인 위치를 나타내는 템플릿을 구성하고 이를 이용하여 영역 최적화와 연결도 최적화를 수행함으로써 빠른 시간 내에 칩 면적의 최소화와 배선장의 최소화를 실현한다. MCNC 빌딩블록 벤치마크 회로에 대해 레이아웃을 수행한 결과, 제안한 계층적 배치 방법이 기존의 방법들에 비하여 수행 시간과 칩 면적 면에서 우수함을 입증한다.

Abstract

In this paper, we propose an algorithm to solve the problem of placement of rectangular blocks whose sizes and shapes are pre-determined. The proposed method solves the placement of many rectangular blocks of different sizes and shapes in a hierarchical manner, so as to minimize the chip area. The placement problem is divided into several sub-problems: hierarchical partitioning, hierarchical area/shape estimation, hierarchical pattern placement, overlap removal, and module rotation. After the circuit is recursively partitioned to build a hierarchy tree, the necessary wiring area and module shapes are estimated using the result of the partitioning and the pin information before the placement is performed. The placement templates are defined to represent the relative positions of the modules. The area and the connectivity are optimized separately at each level of hierarchy using the placement templates, so the minimization of chip area and wire length can be achieved in a short execution time. Experiments are made on the MCNC building block benchmark circuits and the results are compared with those of other published methods. The proposed technique is shown to produce good figures in terms of execution time and chip area.

I. 서론

VLSI의 규모와 복잡도가 증가하여 설계 비용과 시

간이 크게 증가함에 따라, 레이아웃 설계의 자동화가 필요하게 되었다. 레이아웃 설계 방식 중 널리 사용되는 반주문형 설계 방식은 자동화된 설계 도구를 이용하여 빠른 시간 내에 오류가 거의 없는 레이아웃을 설계 방법으로, 다양한 설계 요구조건을 갖는 소량 다품종 IC의 요구가 급격히 증가하면서 그 활용도가 크게 증가하여 특정용도 IC (ASIC)의 대부분을 설계하는 방식으로 자리를 잡고 있다. 이러한 반주문형 설계 방식은 게이트 어레이, 표준셀, 빌딩블록 방식으로 나눌

* 正會員, 建陽大學校 컴퓨터工學科
(Dept. of Computer Eng., Konyang Univ.)

** 正會員, 京畿大學校 電子工學科
(Dept. of Electronics Eng., Kyonggi Univ.)

接受日字: 1996年6月13日, 수정완료일: 1996年10月16日

수 있다

빌딩블록의 배치와 배선 문제는 모양과 크기가 서로 다른 셀 들을 대상으로 레이아웃 설계를 수행하므로 게이트 어레이나 표준셀의 경우보다 설계의 복잡도가 매우 크다.^[11] 빌딩블록의 배치 문제는 가능한 모든 경우에 대해 배치를 수행해야만 최적의 해를 구할 수 있는 조합 최적화 문제이므로 대부분의 배치방법에서 휴리스틱 알고리즘을 사용하고 있다. 기존에 널리 사용되는 방법으로는 min-cut 방법^[2, 31], force-directed 방법^[41], simulated annealing 방법^[51] 등이 있다. Dai^[61]는 계층적 방법을 제안하여, 한번에 5개 이하의 모듈들의 최적 배치를 구하기 위하여 템플리트를 정의 하여 사용했다. Onodera^[71] 등은 배치 문제를 여러 개의 부분으로 분할하여 6개 이하의 모듈의 배치를 branch-and-bound 방법에 의해 최적 배치를 구하는 방법을 사용했다. 이 방법은 배치의 결과가 우수하나 branch-and-bound 방법을 사용하므로 과도한 계산 시간이 요구되어 큰 문제의 배치에는 현실적으로 적용하기 어려운 단점이 있다. 또한, Pan과 Liu^[81]는 cutline 개념을 도입하여 면적 최소화 방법을 사용했으나, 배치 결과에 배선 영역을 포함시키지 않았다.

빌딩블록 배치의 목적함수는 칩 면적의 최소화와 배선장 주, 연결도의 최소화이다. 그러나, 칩 면적과 연결도를 동시에 만족하는 최적화 방법을 찾기는 매우 어렵다.

기존의 min-cut 방법, force-directed 방법, clustering 방법 등은 먼저 연결도 최적화를 수행한 후 면적을 최적화 하는 방법을 사용하므로 게이트 어레이나 표준셀의 배치에 널리 사용되는 방법이다. Simulated annealing 방법은 면적과 연결도를 동시에 최적화 하는 방법으로 2개의 목적함수를 동시에 만족하기 위해 과도한 계산 시간이 소요되는 단점이 있다. 본 논문에서 제안한 계층적 배치방법은 면적 최적화를 수행한 후 연결도 최적화를 수행하는 방법으로 빌딩블록과 같은 임의 크기의 블록을 배치하는데 적합한 방법이다. 특히 빌딩블록은 모듈들의 모양과 크기가 다양하기 때문에 배치과정에서 미사용 영역의 발생을 억제하여 칩 면적을 최소화하는 것이 중요하다.

본 논문에서는 회로를 계층적으로 분할하여 계층 트리를 구성한 후, 분할결과와 각 모듈의 핀 정보를 이용하여 배치가 수행되기 전 단계에서 배선영역과 모듈의 모양을 예측하는 방법을 제안한다. 또한 빌딩블록 레이

아웃 설계의 목적함수인 영역 최적화와 연결도 최적화를 분리하여 수행함으로써, 칩 면적을 감소시키고 수행 속도를 향상시킨 새로운 계층적 배치 방법을 제안한다. 제 2 장에서는 계층적 분할, 계층적 배선영역 예측, 계층적 패턴 배치, min-cut 개념을 이용한 모듈회전, 인접그래프를 이용한 중첩제거의 5 단계로 구성되는 계층적 배치 알고리즘에 대하여 기술한다. 제 3 장에서는 제안된 빌딩블록의 배치 방법을 C언어로 프로그램 하여 실험을 수행하고 이미 발표된 방법들과 결과를 비교함으로써 효용성을 입증한다.

II. 계층구조를 이용한 빌딩블록 배치 방법

본 논문에서 제안한 빌딩블록의 계층적 배치방법은 계층적 트리 분할방법, 계층적 배선영역 예측방법, 계층적 배치방법, min-cut 개념을 이용한 모듈회전 방법, 인접그래프를 이용한 중첩제거 방법의 5단계로 구성된다.

1. C&MC 분할과 계층구조의 구성

본 논문에서는 회로 내에 존재하는 국소적인 cluster를 구성하는데 유리한 bottom-up clustering과, 회로의 연결도를 전체적으로 다루어 분할 결과가 국소최적화에 빠지는 것을 방지할 수 있는 top-down min-cut 분할의 장점을 갖도록, 두 방법을 함께 적용하는 새로운 clustering & min-cut (C&MC) 분할 방법을 제안한다.

(1) C&MC(Clustering & min-cut) 분할

Min-cut 분할 방식^[2, 31]은 회로를 균등한 크기의 2개의 부회로로 초기 분할한 후, 2개의 부회로 내의 모듈을 1 : 1로 교환함으로써 분할을 개선한다. 이 방법은 cut 수가 매우 크고, 비효율적인 분할 결과를 유도할 수도 있다. 이는 회로 내에 존재하는 국소적인 cluster를 고려하지 못한 채, 단지 분할되는 2개의 부회로의 크기를 균등하게 하는 초기 분할을 수행하기 때문에 생기는 단점이다. 특히, 빌딩블록에서는 모듈들의 크기가 크게 다르므로 이를 기준으로 초기 분할을 한 후, 모듈들을 교환하여 분할을 개선하면 결과는 매우 비대칭적인 분할이 될 수 있다.

본 논문에서는 이와 같은 문제점을 해결하기 위하여 clustering에 의해 초기 분할을 수행하고, 이를 min-cut 알고리즘에 의해 개선하는 clustering &

min-cut (C&MC) 분할을 제안한다. C&MC 알고리즘은 clustering에 의한 초기 분할과, 반복적인 clustering 및 min-cut 알고리즘에 의한 분할 개선의 두 단계로 구성된다.

① 초기 분할

회로 내에 존재하는 국소적인 cluster를 처리하고, 분할되는 2개의 부회로의 크기를 가능한 한 균등하게 만들기 위하여 (식-1)과 같은 비용함수를 정의한다. 전체 모듈의 집합을 M, 분할된 2개의 집합을 각각 A와 B라 한다. A, B의 공통 신호선, 즉 cut 수를 CUT_{AB} , 집합 A와 B에 속한 모듈들의 면적의 합을 각각 $Area(A)$, $Area(B)$ 라 할 때, 분할의 비용함수 $\cos t(A, B)$ 는

$$\cos t(A, B) = \frac{CUT_{AB}}{Area(A) \times Area(B)} \quad (1)$$

이다. (식-1)에서 Area항은 분할 결과, 2개의 부회로의 크기를 가능한 균등하게 하기 위한 것이다.

초기 분할은 한번에 한 모듈씩을 선택하면서 (식-1)의 비용함수 값이 최소가 되는 분할 결과를 구하는 과정이다. 우선 다른 모듈들과의 연결도가 가장 작은 모듈을 seed로 선택한다. 다음에 선택될 모듈을 구하기 위하여 나머지 모듈들이 선택될 경우의 비용함수 값을 계산하고, 최소의 값을 유도하는 모듈을 선택한다. 이 과정을 모든 모듈들이 선택될 때까지 반복하는 과정 중에서 최소의 비용함수 값을 갖는 분할을 초기 분할의 결과로 선택한다.

② 분할 개선

초기 분할이 구해지면 그때의 비용함수 값을 저장하고 분할 개선을 시도한다. 초기 분할에서는 집합 B에서 A로 모듈을 이동하였으므로, 분할 개선의 첫 단계에서는 집합 A에서 B로 모듈을 이동하면서 비용함수의 값을 관찰한다. 모든 모듈이 이동하면 그때까지의 비용함수 중 최소인 분할을 새로운 분할 결과로 하여 다시 반대 방향으로(B에서 A로) 이동을 수행한다. 이 과정을 비용함수 값이 감소하지 않을 때까지 반복한다. 분할 개선의 마지막 단계로는 두 집합에서 각각 하나의 모듈을 교환하는 과정이다. 이 단계에서는 기존의 min-cut 알고리즘(FM 알고리즘)을 사용한다.

③ 분할 실험

본 논문에서 제안한 분할 방법의 효용성을 입증하기 위하여 MCNC의 빌딩블록 예제회로에 대한 min-cut

방법과 C&MC 방법의 분할 결과를 표 1에 나타낸다. 표 1에서 cut 수는 분할된 두 집합간의 공통 신호선 수, 모듈 수는 두 집합 내의 모듈 개수, 면적은 두 집합 내에 포함된 모듈들의 면적의 합을 나타낸다. 수행 시간은 SUN4/60(12.5 MIPS)에서의 CPU 시간이다.

표 1에서 C&MC에 의한 분할 결과는 min-cut에 의한 결과에 비해 cut 수가 최대 55.5%, 평균 34.2% 감소됨을 알 수 있다. 분할된 두 집합 A와 B의 면적비(A/B)도 min-cut이 평균 0.66임에 비하여, C&MC는 평균 0.79로 C&MC에 의한 분할 결과가 보다 균등함을 알 수 있다.

다만 수행시간은 회로가 커지면서 큰 차이를 나타낸다. 이는 C&MC 알고리즘에서는 비용함수에 모듈의 크기가 포함되어 있으므로, 하나의 모듈을 clustering 하기 위해 모든 모듈들의 비용함수값을 재계산 해야 하기 때문이다. 그러나, 분할의 수행시간이 전체 레이아웃 수행 시간에서 차지하는 비중은 매우 작기 때문에 우수한 분할 결과를 구하는 것이 더 중요하다.

표 1. 분할 결과의 비교

Table 1. Comparison of partitioning results.

회 로	Min-cut				C&MC			
	cut 수	모듈 수 (A:B)	면적 (mm ²) (A:B)	수행 시간 (초)	cut 수	모듈 수 (A:B)	면적 (mm ²) (A:B)	수행 시간 (초)
xerox	104	4: 6	9.4:10.0	0.10	77	6: 4	9.5: 9.8	1.23
ami33	37	16:17	0.5: 0.6	0.36	27	11:22	0.4: 0.7	54.02
apte	28	4: 5	17.7:29.0	0.13	20	4: 5	23.2:23.5	0.39
ami49	140	9:40	9.2:26.2	0.52	94	10:39	14.9:20.6	240.69
hp	22	5: 6	2.9:5.9	0.12	10	3: 8	3.5: 5.3	0.64

(2) 계층트리의 구성

주어진 회로에 대해 C&MC 알고리즘을 사용하여 순환 반복적으로 분할을 수행함으로써 계층트리를 구성한다.

그림 1에서 계층트리의 루트(root) 노드는 칩 전체를 의미하며, leaf 노드들은 라이브러리에 등록되어 있는 각 모듈들을 나타낸다. 이때, 구성되는 계층트리의 각 노드는 4개 이하의 서브 노드를 포함하도록 한다. 본문에서 제안한 배치 알고리즘의 특성에 의해, 계층트리의 한 노드가 5개 이상의 서브 노드를 갖게 되면 수행시간이 크게 증가하며, 2개 또는 3개의 서브 노드를 갖게 되면 배치 시 모양 불일치(shape mismatch)에

의해 미사용 영역이 과다하게 발생하기 때문이다.¹⁶⁾

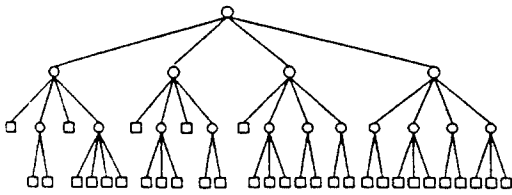


그림 1. 계층트리
Fig. 1. A hierarchy tree.

2. 계층구조를 이용한 배선영역의 예측

모듈과 모듈 사이에 배선영역을 할당하는 문제는 빌딩블록의 배치를 위한 중요한 과정이다. 그러나, 배치가 수행되기 전에 필요한 배선영역을 예측하는 것은 매우 어렵다. 특히, 빌딩블록의 경우에는 각 모듈의 주변에서 배선이 이루어지므로 게이트 어레이나 표준셀보다 배선영역을 예측하기가 어렵다. 즉, 배선영역이 잘못 예측되거나 배선영역이 고려되지 않은 상태로는 배치가 아무리 최적에 가깝게 수행되더라도 배선 시에 삽입되는 배선영역에 의해 배선 후의 결과는 원하는 aspect ratio를 만족할 수 없으며 과다한 미사용 영역에 의해 레이아웃 면적이 크게 증가하게 된다.

본 논문에서는 회로의 분할 결과를 이용하는 새로운 계층적 배선영역 예측 방법을 제안한다.

배선영역은 배치가 수행된 후 개략 배선을 행하면 정확하게 계산될 수 있으나, 배치의 초기 단계에서는 모듈들의 위치 정보가 없기 때문에 배선영역을 정확히 예측할 수 없다. 본 논문에서는 분할 정보를 이용할 수 있도록 계층트리가 구성된 후에 계층구조의 특징을 이용하여 배선영역을 예측한다.

배선영역 예측의 개념을 설명하기 위해 그림 2의 한 노드에 대한 배선영역을 살펴보자. 계층트리의 한 노드에서의 배선영역은 내부 연결도와 외부 연결도에 의해 발생한다. 그림 2에서 노드 A는 4개의 서브노드 1, 2, 3, 4를 갖는다. 따라서, 노드 A의 연결도는 서브노드 1, 2, 3, 4간의 연결도 (이를 노드 A의 내부 연결도라 한다)와 이 4개의 서브 노드들과 A 노드 이외의 노드 (즉, B, C, D)간의 연결도 (이를 노드 A의 외부 연결도라 한다)로 구성된다. 그림 2에서 계층구조의 특징에 의해 현재 레벨에서의 외부 연결도는 상위 레벨에서의 내부 연결도가 됨을 알 수 있다. 내부 연결도에 의한 배선영역을 내부 배선영역, 외부 연결도에 의한 배선영

역을 외부 배선영역이라 한다.

본 논문에서는 현재 레벨에서의 외부 연결도가 상위 레벨에서의 내부 연결도가 되는 계층구조의 특징을 이용하여 배선영역을 예측한다. 내부 배선 영역은 서브노드들의 크기를 알고 있다면 평균 내부 연결도와 평균 신호선 길이에 의해 계산할 수 있다.

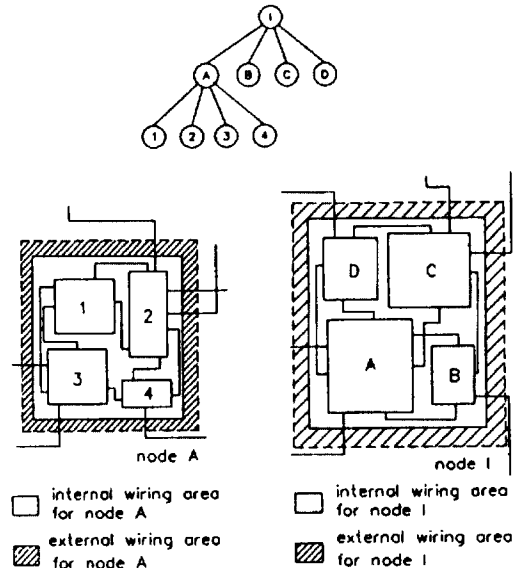


그림 2. 계층구조를 이용한 배선영역 예측의 개념
Fig. 2. The concept of wiring area estimation using hierarchy.

(1) Bottom-up 배선영역 예측

한 노드에서의 배선영역은 내부 배선영역과 외부 배선영역으로 구성된다. 또한, 한 노드에서의 외부 연결도에 의한 외부 배선영역은 상위 레벨의 내부 연결도에 의한 내부 배선영역이 된다.

배선영역 예측의 첫 단계인 bottom-up 배선영역 예측에서는 계층트리를 leaf 노드에서 루트 노드로 순회 하면서 각 노드의 내부 연결도에 의한 배선영역을 예측한다. 최종적으로 루트 노드까지의 모든 노드에서 예측된 배선영역을 합하면 전체 배선영역이 된다. Bottom-up 배선영역 예측 알고리즘은 후위 순회 (post-order traversal) 기법을 사용한다. 즉, bottom-up 배선영역 예측은 leaf 노드에서 루트 노드로 각 노드의 내부 배선영역 예측에 의해 수행된다. 한 노드의 내부 배선영역은 서브 노드들의 크기를 알고 있다면 평균 내부 연결도와 평균 신호선의 길이에 의

해 계산할 수 있다.

k (k ≤ 4)개의 모듈(서브 노드)을 갖는 노드의 내부 배선영역의 계산방법은 다음과 같다. k개의 모듈 사이의 상호 연결도의 총합을 C_T 라 하면,

$$C_T = (\sum_{i=1}^k \sum_{j=1}^k C_{ij})/2 \quad (2)$$

이다. 단, C_{ij} 는 모듈 i와 모듈 j간의 연결도이다.

또한, 모듈 사이의 연결도는 그 사이를 연결하는 모든 신호선의 가중치의 합으로 표현한다. 즉, N_i와 N_j를 각각 모듈 i와 모듈 j에 연결된 신호선의 집합이고, W_n을 신호선 n에 부여된 가중치라 할 때, 모듈 i와 모듈 j간의 연결도, C_{ij} 는

$$C_{ij} = \sum_{n \in N_i \cap N_j} W_n \quad (3)$$

가 된다.

k개의 모듈을 연결하기 위한 평균 신호선의 길이를 L 이라 하면, (식-2) 와 (식-3)으로 부터,

$$\begin{aligned} \text{Estimated Wiring Area} &= t_s \cdot [(\sum_{i=1}^k \sum_{j=1}^k C_{ij})/2] \cdot d_{ij} \\ &= t_s \cdot C_T \cdot L \end{aligned} \quad (4)$$

이다. 단 t_s는 트랙과 트랙 사이의 거리이고 d_{ij}는 모듈 i와 모듈 j의 거리로서, 2 모듈의 중심점 간의 맨해튼 거리이다.

(식-4)에서 t_s는 사용하는 설계 규칙에 의해 결정되는 상수이며, (식-2)에 정의된 C_T는 입력 회로 기술과 분할 결과에 의해 계산 가능하다. 따라서, (식-4)의 예측 배선영역을 계산하기 위해서는 평균 신호선의 길이 L을 구하면 된다. 이때, 각 모듈은 회전이 가능하므로 모듈의 모양을 정사각형으로 가정하고, 핀의 위치는 4 주변의 중심에 위치한다고 가정한다.

(2) Top-down 배선영역 분배

각 모듈에 배선영역이 예측되어 할당되면 배선영역을 포함하는 크기로 모듈을 확장시킨다. Bottom-up 배선영역의 예측이 수행되면 각 모듈은 내부 배선영역이 할당된 확장된 모듈이 된다. 이 확장된 모듈은 외부 연결도에 의해 상위 레벨 모듈의 내부 배선영역을 유도한다. 따라서 그림 3과 같이, top-down 배선영역 분배 단계에서는 모듈의 상위 노드에 할당된 배선영역을 현 노드에 일정한 규칙에 의해 분배한다. 이를 분배된 외부 배선영역이라 한다. 한 노드에 예측된 배선영

역은 top-down 방식으로 각 서브 노드에 분배되고 모듈 면적에 포함하여 확장된 모듈 크기를 갖는다.

Top-down 배선영역 분배 알고리즘은 트리를 루트 노드에서 시작하여, 깊이 우선 탐색 (DFS, Depth First Search) 순으로 각 노드의 예측된 배선영역을 서브 노드로 분배한다. Top down 배선영역 분배에서는 노드의 내부 연결도 (서브 노드간의 연결도)와 각 서브 노드의 크기 (주변길이)에 비례하도록 예측된 한 노드의 내부 배선영역을 각 서브 노드에 분배한다.

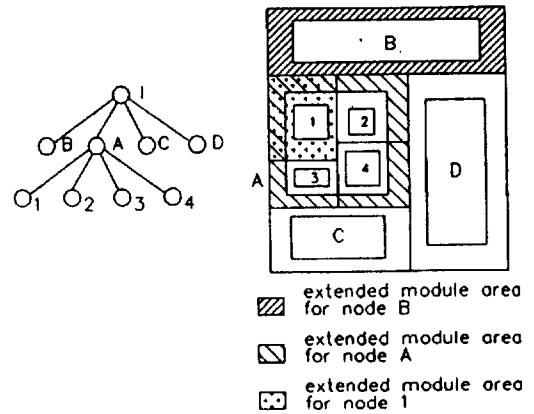


그림 3. Top-down 배선영역 분배
Fig. 3. Top-down wiring area distribution.

(3) Bottom-up 모양 예측

Top-down 방식으로 계층적 배치를 수행하기 위해서는 배선영역의 예측뿐만 아니라 각 노드의 모양을 예측하여야 한다. 이는 한 노드의 배치를 수행하기 위해서는 그 서브 노드들의 영역 뿐 아니라 모양도 결정되어 있어야 하기 때문이다.

본 논문에서는 bottom-up 배선영역 예측과 top-down 배선영역 분배가 끝난 후, 다시 bottom-up 방식으로 모듈모양을 예측한다. 그림 4에서와 같이 노드 A가 4개의 서브 노드 1, 2, 3, 4를 포함한다고 가정한다. 배선영역이 분배된 후에 노드 A의 영역은 서브 노드 1, 2, 3, 4의 영역의 합과 같다. 그러나, 노드 A의 모양은 4개의 서브 노드 1, 2, 3, 4가 배치되는 모양에 따라 크게 달라진다. 본 논문에서는 노드의 모양을 예측하기 위해, 우선, 노드의 모양을 주어진 칩의 aspect ratio와 같이 하고 서브 노드를 포함하지 못하는 경우가 생기면 노드의 모양은 서브 노드의 가장 큰 변을 포함하는 크기를 갖도록 하는 방법을 사용한다.

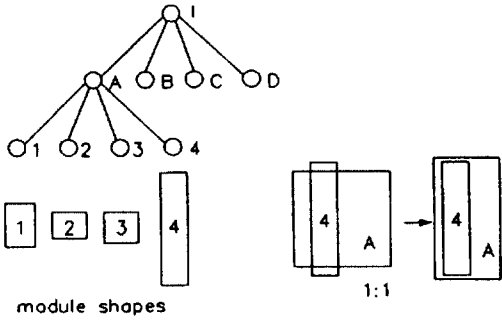


그림 4. Bottom-up 모양 예측
Fig. 4. Bottom-up Shape estimation.

3. 계층적 패턴 배치 알고리즘

(1) 배치 템플리트

빌딩블록의 배치에서는 연결요구의 최적화와 함께 모듈들이 갖는 크기와 모양의 다양성으로 인하여 발생하는 미사용 영역이 최소가 되도록 억제함으로써 칩의 면적을 최소화하는 것이 가장 중요한 목적함수가 된다.

본 논문에서는 템플리트를 이용하여 영역 최적화와 연결도 최적화를 효율적으로 구현하는 top-down 방식의 계층적 패턴 배치 알고리즘을 제안한다. 알고리즘은 루트 노드에서 leaf 레벨로 진행되며, 각 노드에 대해 영역 최적화와 연결도 최적화를 분리하여 수행한다.

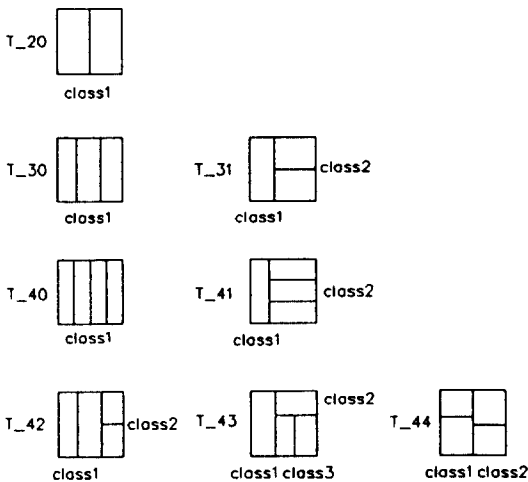


그림 5. 4개 이하의 모듈로 구성되는 템플리트
Fig. 5. Templates with four or less modules.

앞에서 구성한 계층트리는 각 노드가 4개 이하의 서브 노드를 갖는다. 4개 이하의 모듈로 구성 가능한 배치 상태를 모듈간의 상대적 위치관계로 표현하기 위

해 그림 5와 같이 템플리트를 구성하며, 이 템플리트는 2, 3, 4개의 모듈이 배치될 때 발생할 수 있는 모든 상대적인 위치를 표현한다. 따라서 템플리트를 이용하면 배치 문제는 모듈을 각 템플리트의 방으로 할당하는 문제가 된다.

(2) 영역 최적화

영역 최적화 단계에서는 각 템플리트에 대해 최적의 영역을 유도하는 모듈의 조합을 찾는다. 영역 최적화에서는 한 노드가 포함하고 있는 모듈들을 주어진 영역에 가장 잘 맞도록 배치할 수 있는 템플리트와 그 경우의 각 방에 할당되는 모듈의 조합, 그리고 모듈의 회전상태를 결정한다.

그림 5에서 T₃₁의 경우, 모듈이 템플리트의 각 방에 할당되는 경우의 수는 3개의 방에 3개의 모듈을 할당하는 경우의 수이므로 3! = 6가지이다. 그러나, 이때 유도되는 영역을 고려하면 3개의 방은 서로 다른 성질을 갖는 2가지의 클래스로 나누어 생각할 수 있다. 왼쪽 방에 위치할 모듈이 결정되면 나머지 2 모듈은 그 위치에 상관없이 같은 영역을 유도하게 됨을 알 수 있다. 즉, 왼쪽에 있는 하나의 방과 오른쪽에 있는 2개의 방은 서로 다른 클래스로 볼 수 있고, 같은 클래스 내의 방은 모듈의 위치에 상관없이 같은 영역을 유도한다. 따라서 T₃₁의 경우, 3개의 모듈 중 하나를 선택하여 왼쪽 방에 할당하면 나머지 2 모듈은 같은 클래스에 할당되므로, 유도되는 영역의 경우의 수는 ${}_3C_1 = 3$ 이 된다. 또한, 각 모듈의 회전 상태(R0, R90)에 따라 유도되는 영역이 다르므로 T₃₁ 템플리트에 대한 영역 최적화를 위한 총 계산 회수는 ${}_3C_1 * 2^3 = 24$ 이다.

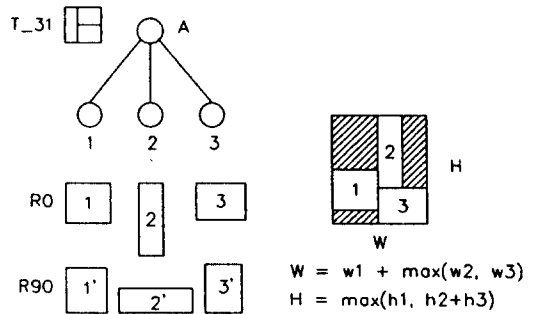


그림 6. 영역 최적화의 비용함수의 계산(T₃₁의 경우)
Fig. 6. Calculation of cost function for area optimization (A case of T₃₁).

각 경우에 템플리트 자체가 90도 회전하는 경우를 포함하여, 모든 경우 중 가장 좋은 비용함수 값을 갖는 조합을 찾는다. 이때의 비용함수는 예측된 모양을 갖는 노드의 영역을 벗어나는 면적으로 정의한다.

비용함수 값의 계산은 각 템플리트에 대하여 미리 저장된 계산 식에 따라 모듈들의 조합을 고려하여 수행된다. 예를 들어, T_31의 비용함수 계산 과정을 그림 6에 나타낸다.

(3) 연결도 최적화

영역 최적화에 의해 템플리트와 모듈의 클래스 및 회전상태(R0, R90)가 결정되면 영역을 변화하지 않도록 각 모듈의 상대적인 위치를 변화하여 연결도를 최적화한다. 한 노드의 연결도 최적화를 위한 목적함수는

$$\text{비용함수} = \sum_{i \in SM} \sum_{j \in M} C_{ij} \cdot d_{ij} \quad (5)$$

의 최소화가 된다. 여기에서 SM(sub-modules)은 현 노드에 포함된 모듈들의 집합이며 M은 회로를 구성하는 전체 모듈의 집합이다. C_{ij} 는 (식-3)에 정의된 모듈 i와 j 사이의 연결도이며, d_{ij} 는 2 모듈 사이의 거리이다.

4. 인접그래프를 이용한 중첩제거 알고리즘

배치 후 모듈간에 중첩이 존재하는 경우, 중첩제거 알고리즘은 중첩이 없는 배치 상태를 구한다. 중첩제거 알고리즘의 목적은 현재의 배치 상태를 가능한 유지하면서 칩 면적의 증가를 최소화하도록 중첩을 제거하는 것이다.

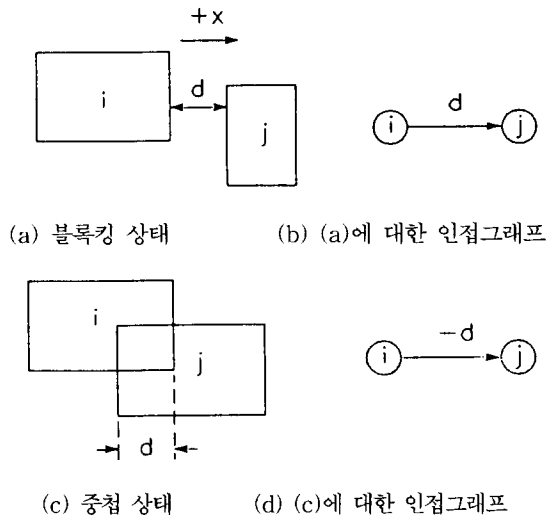


그림 7. 인접관계와 인접그래프
Fig. 7. Adjacency relation and adjacency graph.

본 논문에서는 배치된 모듈간의 상대적인 위치 관계를 정확하게 표현할 수 있는 인접그래프를 이용한 중첩제거 알고리즘을 제안한다. 배치된 각 모듈을 노드로, 각 모듈간의 인접관계를 에지로 표현한 그래프를 인접 그래프(adjacency graph)라 한다. 모듈간의 인접관계는 “중첩”과 “블록킹”에 의해 정의된다. 노드 V_i 가 V_j 에 의해 $+x(+y)$ 방향의 이동이 제한되면 “ V_i 는 V_j 에 의해 $x(y)$ 방향으로 블록킹된다”고 한다. 또한, 이 경우 “ V_j 는 V_i 를 $x(y)$ 방향으로 블록킹한다”고 한다. 그림 7에 블록킹과 중첩에 대한 인접그래프의 표현을 나타낸다.

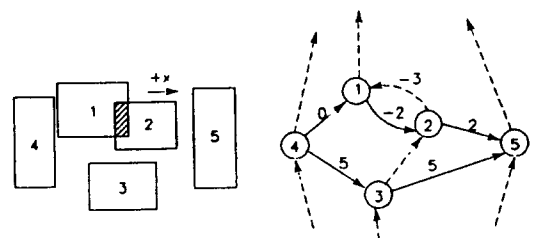
노드 V_i 에서 V_j 로의 인접관계가 존재하고 V_i 에서 V_j 로의 경로가 존재하지 않으면 방향성 에지 (V_i, V_j)를 생성하며 이 에지에는 W_{ij} 의 가중치를 부여한다. 에지의 가중치 W_{ij} 는 V_i 와 V_j 가 중첩되어 있지 않은 경우, V_i 가 이동할 수 있는 최대의 거리를 나타낸다. V_i 와 V_j 가 중첩되어 있는 경우에 W_{ij} 는 음수로 표현되며 크기는 중첩이 제거되기 위해 V_j 가 이동해야 할 최소의 거리를 나타낸다.

인접그래프, G_x, G_y 를 이용한 중첩제거 알고리즘은 그림 8과 같다.

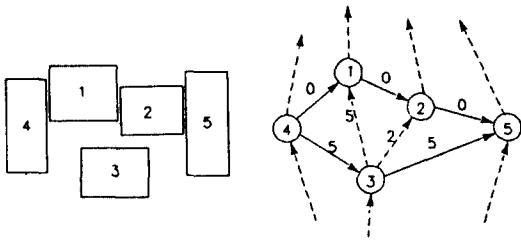
```

Overlap_removal(Adjacency_graphs Gx, Gy)
{
    while(overlap) {
        search_overlapped_node_pair();
        make_overlap_table();
        calculate_overlap_cost();
        remove_overlap_with_smallest_cost();
        update_adjacency_graph(Gx, Gy);
    }
}
    
```

그림 8. 인접그래프를 이용한 중첩제거 알고리즘
Fig. 8. Overlap removal algorithm using adjacency graph.



(a) 이동전의 배치와 이에 대한 인접그래프



(b) 이동후의 배치와 이에 대한 인접그래프

그림 9. 중첩제거 후의 인접그래프의 수정
Fig. 9. Incremental graph update after overlap removal.

중첩제거 과정을 예로 나타내면 그림 9와 같으며, 노드 2가 +x 방향으로 이동할 경우의 수정된 인접 그래프를 나타낸다. y 방향으로 이동할 때에도 같은 방법으로 처리할 수 있다.

5. Min-cut 개념에 의한 모듈회전

배치 과정 중에는 핀의 위치가 모듈의 중심에 위치한다고 가정하고 배치를 수행한다. 따라서, 배치가 끝난 후 모듈들의 배치 상태를 변화시키지 않고 모듈의 회전과 미러링을 수행함으로써 배선 길이를 감소시킬 수 있다. 모듈회전 문제의 목적 함수는 배선 길이의 최소화에 의한 배선 효율의 향상이다.

본 논문에서는 min-cut 개념을 이용한 모듈 회전 알고리즘을 제안한다. 배치된 하나의 모듈에 대하여 중심점을 기준으로 수평과 수직 방향으로 cut-line을 설정하고, min-cut 분할에서와 같은 cut 수에 의한 이득 함수를 계산한다. 이때, 개념적으로 이득함수는 모듈을 미러링하였을 때의 cut 수의 감소량을 의미한다. 즉, 수평 cut-line에 대해 미러링하여 cut 수가 감소하면 수평 방향의 미러링을 수행한다. 같은 방법으로 수직 방향의 미러링 수행 여부를 결정한다. 또한, 모듈을 90도 회전시켜 앞에서와 같이 수평과 수직 cut 수에 의한 이득함수를 계산한다. 모듈 회전의 개념도는 그림 10과 같다.

MCNC^[9]의 빌딩블록 벤치마크 회로 5개에 대해 모듈회전 전후의 배선 길이를 비교한 결과는 표 2와 같다. 배선 길이는 주어진 배치 상태에서 각 모듈에 위치하는 핀의 좌표를 기준으로 minimum spanning tree(MST)를 구한 후, MST의 각 에지의 길이를 맨 해튼 거리로 계산하였다. 실험 결과, 배선 길이가 평균

22.2%가 감소됨을 확인하였다.

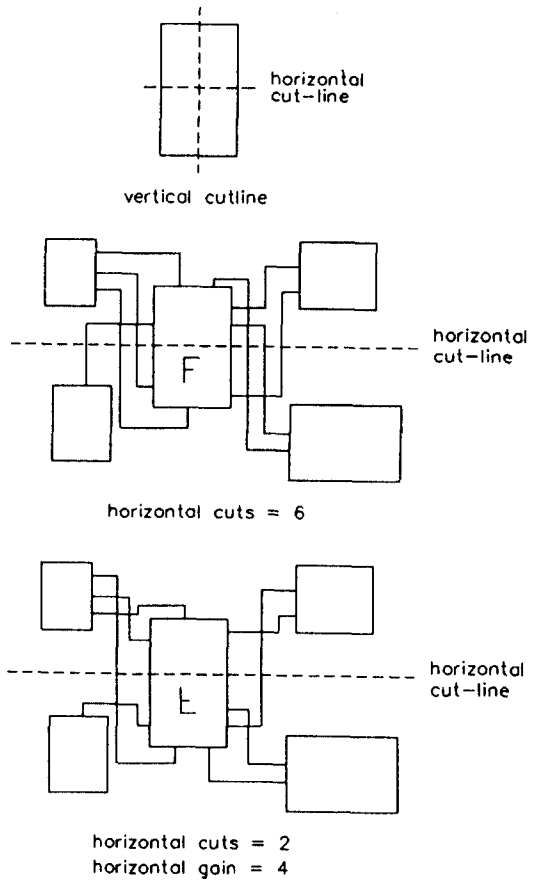


그림 10. 모듈 회전의 개념
Fig. 10. The concepts of module rotation.

표 2. 모듈회전 전후의 배선장(μm)
Table 2. Wire length of circuits before and after module rotation(μm).

회로명	모듈회전 전 배선장 (μm)	모듈회전 후 배선장 (μm)	개선률 (%)
xerox(BBL1)	1307015	1175703	10.0%
ami33(BBL2)	211958	195222	8.9%
apte(BBL3)	507286	259608	48.8%
ami49(BBL4)	2722598	2443470	10.3%
hp(BBL5)	203562	136618	32.9%

Ⅲ. 실험 및 고찰

빌딩블록 레이아웃은 사용하는 모듈의 크기와 모양

이 다양하고, 배선영역이 일정하지 않으므로 배치만으로 그 성능을 평가할 수 없다. 따라서 일반적으로 배선까지 수행한 시스템의 최종 결과로 배치 알고리즘의 성능을 평가한다.^[6, 10-16] 본 논문에서 제안한 빌딩블록 배치 방법의 효율성을 평가하기 위하여, 개략 배선기 및 상세 배선기^[17]와 결합하여 빌딩블록 레이아웃 시스템을 구성하고 벤치마크 회로에 대한 실험을 수행하였다.

본 논문의 빌딩블록 레이아웃 시스템은 C언어로 프로그래밍 되어 SUN/4 워크스테이션 상에 구현하였다. 본 논문에서 제안한 새로운 배치방법의 성능을 평가하기 위하여 1990년 MCNC^[9]에서 제공한 5개의 벤치마크 데이터에 대한 실험을 수행하였다.

표 3에서는 벤치마크 회로에 대하여 본 논문의 시스템을 사용하여 레이아웃 결과를 나타낸다. 실험에서 모든 예제는 100% 배선이 완료되었으며, 면적은 배치 및 배선이 끝난 후 compaction^[18]을 수행한 결과이다. 수행속도는 입력으로부터 배치 결과의 생성까지 소요되는 시간을 나타낸 것이며, 이는 SUN/4-60 (12.5 MIPS, 1.4 MFLOPS)에서의 CPU 시간이다.

표 3. 빌딩블록 레이아웃 시스템의 레이아웃 실험 결과

Table 3. Layout results of building block layout system.

회로명	면적 [mm ²]	배선장 [μm]	via 수 [개]	CPU 시간 [초]
xerox(BBL1)	27.19	692696	1247	23.6
ami33(BBL2)	2.35	116654	848	120.4
apte(BBL3)	51.79	242222	291	17.6
ami49(BBL4)	55.58	1110532	2179	431.7
hp(BBL5)	12.43	245186	421	28.5

표 4와 5에서 벤치마크 데이터로 널리 사용되고 있는 xerox와 ami33에 대한 레이아웃 결과를 이미 발표된 시스템들의 결과와 비교하였다. 표에서 기종 및 수행시간은 레이아웃을 수행한 컴퓨터의 기종과 배치에 소요된 시간을 나타낸다.

표 4와 5의 실험에 사용된 각 시스템의 배치 알고리즘을 살펴보면 다음과 같다. BEAR^[6], FRODO^[14], GRCA^[15] 등은 휴리스틱 알고리즘을 사용하였으며 MOSAICO^[10], VITAL^[11], Delft^[12], Seattle Silicon^[13] 등은 simulated annealing 알고리즘을 사

용하였다. 또한, Minnesota^[16]는 mixed integer programming을 사용했으며, BB^[7]는 branch-and-bound 알고리즘을 사용했다. 알고리즘의 구현 방법에 따라 차이가 있지만 일반적으로 휴리스틱 알고리즘은 계층구조를 사용하여 수행속도가 빠른 반면, simulated annealing이나 branch-and-bound 알고리즘은 수행속도는 길지만 결과가 비교적 우수하다.

표 4. Xerox에 대한 레이아웃 결과 비교

Table 4. Comparison of layout results on xerox.

시스템	면적 (mm ²)	배선장 (μm)	via 수 (개)	기종	수행시간 (초)
MOSAICO	29.01	650009	1173	-	-
VITAL	31.71	865712	1029	-	-
BEAR[6]	28.47	633494	897	VAX8800	131
Delft[12]	34.31	1039832	1429	-	-
Seattle Silicon[13]	25.79	601300	1104	DN4000	540
FRODO[14]	29.06	-	-	SUN4	28
GRCA[15]	28.80	-	-	-	-
BB[7]	26.17	628000	-	DEC3100	4,001
본 논문	27.19	692696	1247	SUN4/60	23.6

* 참고문헌이 명시되지 않은 데이터는 BEAR^[6]의 데이터임.

표 5. Ami33에 대한 레이아웃 결과 비교

Table 5. Comparison of layout results on ami33.

시스템	면적 (mm ²)	배선장 (μm)	via 수 (개)	기종	수행시간 (초)
MOSAICO	3.16	151824	813	-	-
VITAL	3.12	134599	763	-	-
BEAR[6]	2.83	131244	798	VAX8800	762
Delft[12]	2.97	160855	962	-	-
Seattle Silicon[13]	2.42	91100	654	DN4000	2,940
FRODO[14]	2.92	-	-	SUN4	50
GRCA[15]	2.88	-	-	-	-
Minnesota[16]	2.54	138910	-	DN3550	360
BB[7]	2.24	109000	-	DEC3100	5,344
본 논문	2.35	116654	848	SUN4/60	120.4

* 참고문헌이 명시되지 않은 데이터는 BEAR^[6]의 데이터임.

특히 BB¹⁷⁾에서는 branch-and-bound 알고리즘을 사용하여 전체 해법 중 최적의 해를 탐색하는 방법 즉, 모듈들의 상대적인 위치와 회전 상태를 바꾸면서 가능한 모든 해를 탐색하기 함으로써 배치 결과는 가장 좋지만 수행 속도가 과다하게 소요된다. 예를 들어, 10개의 모듈로 구성된 xerox의 배치 시간은 DEC-3100(16.2 MIPS)에서 294분이 소요되었으며, 배선까지 포함하면 약 40시간이 소요된다.¹⁷⁾ 또한, 배치 수행 시간을 감소시키기 위하여 계층적으로 배치를 수행하는 경우에도 xerox 회로에 대해 4,001초, ami33 회로에 대해 5,344초가 소요되어 회로의 규모가 커짐에 따라 branch-and-bound 알고리즘을 적용하기에는 어려움이 있음을 알 수 있다. 반면 본 논문의 방법은 xerox 회로의 경우 23.6초, ami33 회로에 대해 120.4초가 소요되어 회로의 규모가 커짐에 따라 수행시간이 크게 감소됨을 알 수 있다. 실험 결과로부터, 본 논문의 시스템은 기존에 발표된 시스템들에 비하여 레이아웃 면적과 설계 수행속도 면에서 매우 우수한 결과를 보였으며, 이는 영역과 연결도를 분리하여 최적화 하는 배치 알고리즘의 특징에 기인한다.

그림 11, 12는 본 시스템을 사용하여 xerox, ami33 회로에 대해 레이아웃을 수행한 결과를 나타낸다.

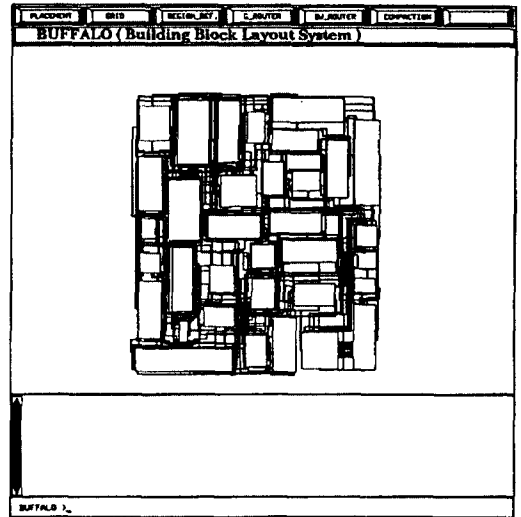


그림 12. Ami33 회로의 레이아웃 결과
Fig. 12. Layout result of ami33 benchmark data.

IV. 결 론

본 논문에서는 빌딩블록의 레이아웃 설계 자동화를 위한 새로운 계층적 배치 방법을 제안하였다. 제안된 빌딩블록의 새로운 계층적 배치 방법은 계층구조의 구성을 위한 분할, 배선영역 예측, 계층적 배치, 중첩 제거, 모듈 회전의 5 단계로 구성하였다.

새로운 clustering/min-cut(C&MC) 분할 알고리즘에 의해 회로를 반복적으로 분할하여 계층구조를 구성하고 이를 계층 트리로 표현하였다. 실험 결과, C&MC 방법은 기존의 min-cut 분할에 비해 cut 수가 평균 34.2% 감소하였으며 분할되는 부회로의 크기도 더 균등하였다. 회로의 분할 결과와 각 모듈의 핀 정보를 이용하는 계층적 배선영역 예측 알고리즘을 제안하여, 배치가 수행되기 전 단계에서 각 모듈의 주변에 배선 영역을 할당하고 모듈의 모양을 결정하였다. 계층적 배치 단계에서는 빌딩블록 배치의 목적함수인 영역 최적화와 연결도 최적화를 분리하여 수행하는 새로운 알고리즘을 제안하였다. 이 새로운 방법은 영역 최적화와 연결도 최적화를 분리하여 수행함으로써 짧은 시간 내에 가능한 배치를 모두 탐색하여 최적의 해를 구할 수 있다. 배치 후 중첩이 존재하는 경우, 인접 그래프를 이용하여 현재까지의 배치 결과를 최대한 유지하고 침면적의 증가를 최소화하는 새로운 중첩제거 알고리즘을 적용하였다. 또한, 중첩제거 후에 존재하는 미사용

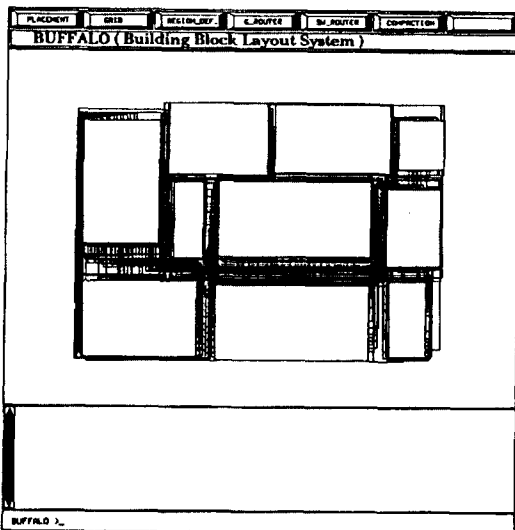


그림 11. Xerox 회로의 레이아웃 결과
Fig. 11. Layout result of xerox benchmark data.

영역을 배선 시 배선 영역으로 활용할 수 있도록 모듈 주변에 균일하게 분배한다. 또한, 배선 효율을 향상시키기 위해 min-cut 개념을 확장시킨 새로운 모듈회전 알고리즘을 제안하였다. 실험 결과 minimum spanning tree로 계산한 핀간의 거리가 모듈회전 후 평균 22.2% 향상됨을 확인하였다.

본 논문에서 제안된 빌딩블록 배치 방법은 개략 배선기, 상세 배선기 및 compactor와 결합하여 레이아웃 시스템을 구성하였다. 벤치마크 회로에 대하여 레이아웃을 수행하고 기존에 발표된 시스템과 비교한 결과, 레이아웃 면적 및 설계 수행속도가 매우 우수함을 확인하였다. 특히 회로의 규모가 증가하여 수행 속도가 중대한 문제가 되는 경우, 본 논문의 빌딩블록 배치 방법은 빠른 시간 내에 우수한 레이아웃을 자동 설계하여 설계 비용을 절감할 수 있을 것으로 기대된다.

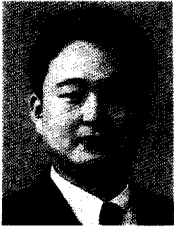
참 고 문 헌

- [1] Kuh, E. S. and T. Ohtsuki, "Recent Advances in VLSI Layout," Proceedings of the IEEE, Vol. 78, No.2, pp.237-263, Feb. 1990.
- [2] Fiduccia, C. M. and R. M. Mattheyses, "A Linear-Time Heuristic for Improving Network Partitions", Proc. of 19th Design Automation Conf., pp.175-181, 1982.
- [3] Kernighan, B. W., S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs", Bell System Technical Journal, Vol.49, pp.291-308, Feb. 1970.
- [4] Quinn, N. R. Jr. and M. A. Breuer, "A Force Directed Component Placement Procedure for Printed Circuit Board", IEEE Trans. on Circuit and System, Vol.CAS-26, pp. 377-388, Jun. 1979.
- [5] Kirkpatrick, S., C. D. Gelatt, Jr. and M. P. Vecchi, "Optimization by Simulated Annealing", Science, Vol.220, pp.671-680, May. 1983.
- [6] Dai, W. M., B. Eschermann, E. S. Kuh, M. Pedram, "Hierarchical Placement and Floorplanning in BEAR," IEEE Trans. on Computer Aided Design, Vol.8, No.12, pp.1335-1349, Dec. 1989.
- [7] Onodera, H., Y. Taniguchi, K. Tamaru, "Branch-and-bound Placement for Building Block Layout," 28th Design Automation Conf., pp.433-439, 1991.
- [8] Pan, P and Liu, C. L, "Area Minimization for floorplans," IEEE Trans. on Computer Aided Design, No.1, Jan, 1995.
- [9] Preas, B and Robert, K., Physical Design Workshop, Hilton Head, South Carolina, 1987.
- [10] Sangiovanni-Vincentelli, A., et. al, "Mosaico: An Integrated Macro-Cell Layout System," International Workshop on Placement & Routing, May. 1988.
- [11] Smith, D. C., et. al., "VITAL: A Macro, Standard-cell, and Gate-array Chip Compiler," International Workshop on Placement & Routing, May, 1988.
- [12] Cai, H., "Connectivity Biased Channel Construction and Ordering for Building Block Layout," Proc. of 25th Design Automation Conf., pp.560-565, 1988.
- [13] Upton, M., K. Samii, S. Sugiyama, "Integrated Placement for Mixed Macro Cell and Standard Cell Design," Proc. of 27th Design Automation Conf., pp.32-35, 1990.
- [14] Lengauer, T., R. Muller, "A Robust Framework for Hierarchical Floorplanning With Integrated Global Wiring," Proc. of ICCAD-90, pp.148-151, 1990.
- [15] Herrigel, A., "GRCA: A Global Approach for Floorplanning Synthesis in VLSI Macrocell Design," Proc. of ICCAD-90, pp.152-155, 1990.
- [16] Sutanthavibul, S., E. Shragowitz, "An Analytic Approach to Floorplan Design and Optimization," Trans. on Computer Aided Design, Vol.10, No.6, pp.761-769, Jun. 1991.
- [17] Seung-Ho Lee, Byung-Ik Kang, Jong-Wha Chong and Seung-Youn Kim, "Dinosaur : A New General Multi-layer Area Router," The AOM 13th Annual International Conference, Proc. of the Computer Science Group, vol.13, No.1, pp.226-229, British Columbia, Canada, Aug.2-5, 1995.
- [18] Shin, H., A. Sangiovanni-Vincentelli, and C. Sequin, "Two-dimensional Compaction by

zone refining," Proc. of 23rd Design

Automation Conf., pp.115-122, Jun.1986.

저 자 소 개



姜 乘 益(正會員)

1961년 11월 22일 생. 1984년 2월 한양대학교 전자공학과 졸업(공학사). 1986년 2월 한양대학교 대학원 전자공학과 졸업(공학석사). 1992년 8월 한양대학교 대학원 전자공학과 졸업(공학박사). 1991년 3월 - 현재

한양대학교 컴퓨터공학과 조교수. 주관심분야는 VLSI CAD, GIS 등임



李 建 培(正會員)

1959년 8월 20일 생. 1982년 2월 한양대학교 전자공학과 졸업(공학사). 1984년 2월 한양대학교 대학원 전자공학과 졸업(공학석사). 1989년 2월 한양대학교 대학원 전자공학과 졸업(공학박사). 1991년 9월 - 현재

경기대학교 전자공학과 조교수. 주관심분야는 VLSI 설계, GIS, 멀티미디어